UNCLASSIFIED

CENTRAL RESEARCH LIBRARY
DOCUMENT COLLECTION

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES
3 4456 0350094 0

ORNL-2047
Physics

cy. 4

A THREE-REGION, TWO-GROUP, TWO-DIMENSIONAL REACTOR CODE FOR THE ORACLE

C. L. Bradshaw

CENTRAL RESEARCH LIBRARY
DOCUMENT COLLECTION
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this document, send in name with document and the library will arrange a loan.

OAK RIDGE NATIONAL LABORATORY
OPERATED BY
UNION CARBIDE NUCLEAR COMPANY
A Division of Union Carbide and Carbon Corporation
UCC
POST OFFICE BOX P · OAK RIDGE, TENNESSEE

UNCLASSIFIED

ORNL 2047

Contract No. W-7405-eng-26

MATHEMATICS PANEL

A THREE-REGION, TWO-GROUP, TWO-DIMENSIONAL REACTOR CODE FOR THE ORACLE

C. L. Bradshaw

DATE ISSUED

MAR 1 5 1956

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0350094 0

OAK RIDGE NATIONAL LABORATORY
Operated by
UNION CARBIDE NUCLEAR COMPANY
A Division of Union Carbide and Carbon Corporation
Post Office Box P
Oak Ridge, Tennessee

INTERNAL DISTRIBUTION

1. C. E. Center
2. Biology Library
3. Health Physics Library
4-5. Central Research Library
6. Reactor Experimental Engineering Library
7-26. Laboratory Records Department
27. Laboratory Records, ORNL R.C.
28. A. M. Weinberg
29. L. B. Emlet (K-25)
30. J. P. Murray (Y-12)
31. J. A. Swartout
32. E. H. Taylor
33. E. D. Shipley
34. A. S. Householder
35. F. C. VonderLage
36. C. P. Keim
37. W. H. Jordan
38. S. C. Lind
39. F. L. Culler
40. R. S. Livingston
41. A. H. Snell
42. A. Hollaender
43. M. T. Kelley
44. G. H. Clewett
45. J. H. Frye, Jr.
46. K. Z. Morgan
47. T. A. Lincoln
48. C. S. Harrill
49. D. W. Cardwell
50. C. E. Winters
51. E. M. King
52. Lewis Nelson
53. D. D. Cowen
54. J. A. Lane
55. R. A. Charpie
56. S. J. Cromer
57. M. J. Skinner
58. R. R. Dickison
59. J. H. Alexander
60. R. R. Bate
61. H. C. Claiborne
62. R. R. Coveyou
63. N. M. Dismuke
64. A. C. Downing
65. T. B. Fowler
66. C. L. Gerberich
67. P. R. Kasten
68. W. E. Kinney
69. J. Niestlie
70. E. Silver
71. A. Simon
72. C. D. Zerby
73-97. B. J. Osborne
98. G. C. Williams
99. ORNL - Y-12 Technical Library, Document Reference Section

EXTERNAL DISTRIBUTION

100-105. C. L. Bradshaw, Redstone Arsenal, Huntsville, Alabama
106. W. R. Mann, University of North Carolina, Chapel Hill, North Carolina
107. W. C. Sangren, Curtiss-Wright Corp., Clifton, New Jersey
108. Grady Cox, Armanent Testing Center, Eglin AFB, Eglin Field, Florida
109. Jack Chernick, Brookhaven National Laboratory
110. D. A. Flanders, Argonne National Laboratory
111. Jack Heller, AEC Computing Facility, New York University
112. Herb Keller, AEC Computing Facility, New York University
113. Ben Mount, Westinghouse Electric Corp., Pittsburgh, Pennsylvania
114. Richard Stark, Knolls Atomic Power Laboratory
115. R. F. Bacher, California Institute of Technology
116. Division of Research and Development, AEC, ORO
117-751. Given distribution as shown in TID-4500 under Physics category (300 copies - OTS)

DISTRIBUTION PAGE TO BE REMOVED IF REPORT IS GIVEN PUBLIC DISTRIBUTION

TABLE OF CONTENTS

Summary 1

Mathematical Description of the Problem. 1

Method of Solution 4

The Difference System. 7

The Code 15

Preparation of Input Data. 22

The Problem Output 28

Limitations of the Code. 29

Operating Procedure. 29

Nomenclature 32

SUMMARY

The code described in this report has served as an intermediate step to the multi-region, multi-group, two-dimensional reactor calculation. The particular geometry of the code was chosen since there was an immediate need for results from a code of this type. The problem solved is, in general, a three region cylindrical problem using two groups of neutrons. Provision is also made for a thin shell surrounding the core. This shell is considered to be transparent to fast neutrons and an absorber of thermal neutrons. The code can be run with or without the shell and as a two-region problem if so desired. The code has been made as general as possible so that it may be applied to any problem which can be suitably approximated by a two-group three-region cylindrical model.

MATHEMATICAL DESCRIPTION OF THE PROBLEM

The problem solved by the code is indicated by the following figure, differential equations and boundary conditions:

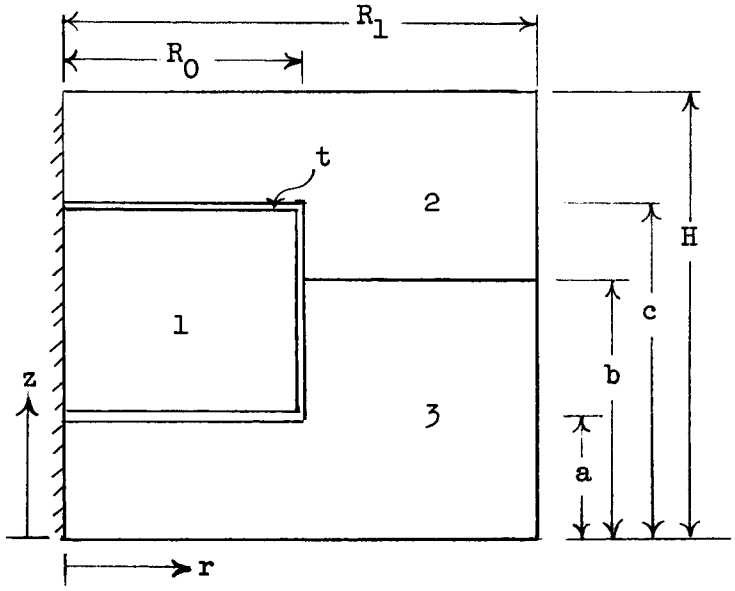

FIGURE 1

The figure indicates that cylindrical coordinates (r , z) are used and that symmetry is assumed about the z-axis.

Differential equations:

(1) $$D_{f_i} \nabla^2 \varphi_{f_i} + \frac{k_i}{p_i} \Sigma_{s_i} \varphi_{s_i} - \Sigma_{f_i} \varphi_{f_i} = 0$$

(2) $$D_{s_i} \nabla^2 \varphi_{s_i} + p_i \Sigma_{f_i} \varphi_{f_i} - \Sigma_{s_i} \varphi_{s_i} = 0$$

i = 1, 2, 3 refers to the three regions, s and f to the two neutron groups, slow and fast.

Boundary conditions:

(3.a) $$\left(\frac{\partial \varphi_{j_i}}{\partial r}\right)_{r=0} = 0 \quad ; \qquad i=1,2,3;\ j=f,s\ .$$

(3.b) $$\varphi_{j_i}(r = R_1, z) = 0 \quad ; \qquad i=2,3\ ;\ j=f,\ s\ .$$

(3.c) $$\varphi_{j_3}(r,\ z = 0) = 0 \quad ; \qquad j = f,\ s\ .$$

(3.d) $$\varphi_{j_2}(r,\ z = H) = 0 \quad ; \qquad j = f,\ s\ .$$

(3.e) $$\varphi_{j_2}(r > R_o,\ z = b) = \varphi_{j_3}(r > R_o,\ z = b) \quad ; \qquad j = f,\ s\ .$$

(3.f) $$D_{j_2} \left(\frac{\partial \varphi_{j_2}}{\partial z}\right)_{r > R_o, z=b} = D_{j_3} \left(\frac{\partial \varphi_{j_3}}{\partial z}\right)_{r > R_o, z=b} \quad ; \quad j = f,\ s\ .$$

(3.g) $$\varphi_{j_2}(r < R_o,\ z = c) = \varphi_{j_1}(r < R_o,\ z = c) \quad ; \qquad j = f,\ s \ .$$

(3.h) $$D_{f_2}\left(\frac{\partial\varphi_{f_2}}{\partial z}\right)_{r < R_o, z=c} = D_{f_1}\left(\frac{\partial\varphi_{f_1}}{\partial z}\right)_{r < R_o,\ z=c}$$

(3.i) $$D_{s_2}\left(\frac{\partial\varphi_{s_2}}{\partial z}\right)_{r < R_o, z=c} = D_{s_1}\left(\frac{\partial\varphi_{s_1}}{\partial z}\right)_{r < R_o, z=c} + \Sigma_a\, t\, \varphi_{s_1}(r < R_o,\ z = c)$$

(3.j) $$\varphi_{j_3}(r < R_o,\ z = a) = \varphi_{j_1}(r < R_o,\ z = a) \quad ; \qquad j = f,\ s$$

(3.k) $$D_{f_3}\left(\frac{\partial\varphi_{f_3}}{\partial z}\right)_{r < R_o,\ z=a} = D_{f_1}\left(\frac{\partial\varphi_{f_1}}{\partial z}\right)_{r < R_o,\ z = a}$$

(3.ℓ) $$D_{s_3}\left(\frac{\partial\varphi_{s_3}}{\partial z}\right)_{r < R_o, z=a} = D_{s_1}\left(\frac{\partial\varphi_{s_1}}{\partial z}\right)_{r < R_o, z=a} - \Sigma_a\, t\, \varphi_{s_1}(r < R_o, z=a)$$

(3.m) $$\varphi_{j_1}(R_o,\ b < z < c) = \varphi_{j_2}(R_o,\ b < z < c) \quad ; \qquad j = f,\ s$$

(3.n) $$\varphi_{j_1}(R_o,\ a < z < b) = \varphi_{j_3}(R_o,\ a < z < b) \quad ; \qquad j = f,\ s$$

$$(3.o) \qquad D_{f_1}\left(\frac{\partial\varphi_{f_1}}{\partial r}\right)_{r=R_o, b<z<c} = D_{f_2}\left(\frac{\partial\varphi_{f_2}}{\partial r}\right)_{r=R_o, b<z<c}$$

$$(3.p) \qquad D_{f_1}\left(\frac{\partial\varphi_{f_1}}{\partial r}\right)_{r=R_o, a<z<b} = D_{f_3}\left(\frac{\partial\varphi_{f_3}}{\partial r}\right)_{r=R_o, a<z<b}$$

$$(3.q) \qquad -D_{s_1}\left(\frac{\partial\varphi_{s_1}}{\partial r}\right)_{R_o, b<z<c} = D_{s_2}\left(\frac{\partial\varphi_{s_2}}{\partial r}\right)_{R_o, b<z<c} + \Sigma_a\, t\, \varphi_{s_1}(R_o,\ b<z<c)$$

$$(3.r) \qquad -D_{s_1}\left(\frac{\partial\varphi_{s_1}}{\partial r}\right)_{R_o, a<z<b} = -D_{s_3}\left(\frac{\partial\varphi_{s_3}}{\partial r}\right)_{R_o, a<z<b} + \Sigma_a\, t\, \varphi_{s_1}(R_o,\ a<z<b)\ .$$

METHOD OF SOLUTION

It is assumed that the material make-up of the three regions is known and that one is interested in finding the effective multiplication constant for the given reactor system. In order to produce this number k_{eff}, (which will in all subsequent references be called simply K) , the set of differential equations is replaced by a finite difference system and this difference system is solved numerically. The numerical method requires the solution of two large systems of algebraic equations (one for each of the two groups) several times to gain the required convergence to K . An outline of the determination of K is as follows:

A. The differential system is replaced by a difference system using approximations of the form,

$$\varphi_{zz} = \frac{1}{h^2} (\varphi_{m+1,n} - 2\varphi_{m,n} + \varphi_{m-1,n}) \quad \text{etc.}$$

The boundary conditions are carried over and applied to the difference system. Mesh points which lie on interfaces, in particular corner points, are the most difficult to approximate. For this reason a somewhat different mesh system is used in which no mesh points are allowed to fall along interfaces. This mesh was suggested by A. C. Downing and has the property that all interfaces are located midway between adjacent lines of the lattice of mesh points. This is discussed in more detail in a subsequent section.

B. Taking the steps as indicated in (A), two systems of algebraic equations are obtained. In matrix notation these take the form,

(4) $$A_f \varphi^f + F_f \varphi^s = 0$$

(5) $$A_s \varphi^s + F_s \varphi^f = 0 \quad .$$

A_f and A_s are square matrices, the order of both being equal to the number of mesh points used $\left[I\right]$. F_f and F_s are diagonal matrices with order the same as A_f and A_s . φ^f and φ^s are the fast and slow flux vectors (one component for each point in the mesh).

C. The iterative scheme used to solve the system (4) and (5) is as follows:

1. A first estimate for φ_o^s and K_o is made. From (4) we then have a system of the form,

(6) $$A_f \varphi^f + D_o = 0 \quad .$$

2. φ_o^f is obtained from the system (6) using the iterative method of Richardson. The recursive formula of Richardson can be written as

(7) $$\varphi_i^{n+1} = \varphi_i^n + \beta_n \left\{ \sum_{j=1}^{N} a_{ij} \varphi_j^n + D_i^n \right\} \quad , \quad i = 1, 2, \ldots, N \ ,$$

where the β_n are properly chosen relaxation factors (factors chosen to increase the rate of convergence).

3. φ_o^f is inserted into the system (5) with the result

$$A_s \varphi_1^s + D_1 = 0 \quad ,$$

and this system is solved for φ_1^s .

4. A new estimate of K is obtained from

$$K_{n+1} = \frac{\varphi_{n+1}^s}{\varphi_n^s / K_n} \quad .$$

The vector φ^n_{n+1} is normalized by dividing by K_{n+1} and the cycle is reentered at (1). This is repeated until the desired convergence to K is complete.

When the evaluation of K has been completed, the code then computes the following integrals in order to obtain a neutron balance.

(8.a) $$A^T_{s_i} = \Sigma_{s_i} \int_V \varphi_s \, dV \quad ; \qquad i = 1, 2, 3 \; .$$

(8.b) $$A^T_{f_i} = \Sigma_{f_i} (1 - p_i) \int_V \varphi_f \, dV \quad ; \qquad i = 1, 2, 3 \; .$$

(8.c) $$A^T_{s_i}(\text{shell}) = V_{\text{shell}} \cdot \Sigma_a \cdot \varphi_s \quad ; \qquad i = 2, 3 \; .$$

(9.a) $$L^r_{f_i} = -D_{f_i} \int_S \left(\frac{\partial \varphi_{f_i}}{\partial n}\right)_r \cdot dS \quad ; \qquad i = 2, 3 \; .$$

(9.b) $$L^r_{s_i} = -D_{s_i} \int_S \left(\frac{\partial \varphi_{s_i}}{\partial n}\right)_r \cdot dS \quad ; \qquad i = 2, 3 \; .$$

(9.c) $$L^z_{f_i} = -D_{f_i} \int_S \left(\frac{\partial \varphi_{f_i}}{\partial n}\right)_z \cdot dS \quad ; \qquad i = 2, 3 \; .$$

(9.d) $$L^z_{s_i} = -D_{s_i} \int_S \left(\frac{\partial \varphi_{s_i}}{\partial n}\right)_z \cdot dS \quad ; \qquad i = 2, 3 \; .$$

THE DIFFERENCE SYSTEM

In cylindrical coordinates (r - z)

$$\nabla^2 \varphi = \frac{\partial^2 \varphi}{\partial r^2} + \frac{1}{r} \frac{\partial \varphi}{\partial r} + \frac{\partial^2 \varphi}{\partial z^2} \; .$$

Using a second order approximation for the derivative, we have

$$\nabla^2 \varphi \simeq \frac{\varphi_{m+1,n} - 2\varphi_{m,n} + \varphi_{m-1,n}}{(\Delta r)^2} + \frac{1}{r_m} \frac{\varphi_{m+1,n} - \varphi_{m-1,n}}{2\Delta r} + \frac{\varphi_{m,n+1} - 2\varphi_{m,n} + \varphi_{m,n-1}}{(\Delta z)^2} \quad . \tag{10}$$

Making this approximation in equation (1) and (2), the following difference equations result:

$$D_{f_i}\left[1 + \frac{\Delta r}{2r_m}\right]\varphi^{f_i}_{m+1,n} + D_{f_i}\left[1 - \frac{\Delta r}{2r_m}\right]\varphi^{f_i}_{m-1,n} + \left\{D_{f_i}\left[-2-2\left(\frac{\Delta r}{\Delta z}\right)^2\right] - (\Delta r)^2\Sigma_{f_i}\right\}\varphi^{f_i}_{m,n}$$

$$+ D_{f_i}\left(\frac{\Delta r}{\Delta z}\right)^2 \varphi^{f_i}_{m,n+1} + D_{f_i}\left(\frac{\Delta r}{\Delta z}\right)^2 \varphi^{f_i}_{m,n-1} + (\Delta r)^2 \frac{k_i}{p_i}\Sigma_{s_i}\varphi^{s_i}_{m,n} = 0 \quad , \tag{11}$$

$$D_{s_i}\left[1 + \frac{\Delta r}{2r_m}\right]\varphi^{s_i}_{m+1,n} + D_{s_i}\left[1 - \frac{\Delta r}{2r_m}\right]\varphi^{s_i}_{m-1,n} + \left\{D_{s_i}\left[-2-2\left(\frac{\Delta r}{\Delta z}\right)^2\right] - (\Delta r)^2\Sigma_{s_i}\right\}\varphi^{s_i}_{m,n}$$

$$+ D_{s_i}\left(\frac{\Delta r}{\Delta z}\right)^2 \varphi^{s_i}_{m,n+1} + D_{s_i}\left(\frac{\Delta r}{\Delta z}\right)^2 \varphi^{s_i}_{m,n-1} + (\Delta r)^2 p_i \Sigma_{f_i}\varphi^{f_i}_{m,n} = 0 \quad . \tag{12}$$

The general case, in so far as multi-regions are concerned, is considered in the application of the boundary conditions. The mesh scheme employed has already been mentioned and its application to the boundary conditions will be described with the aid of Figure 2.

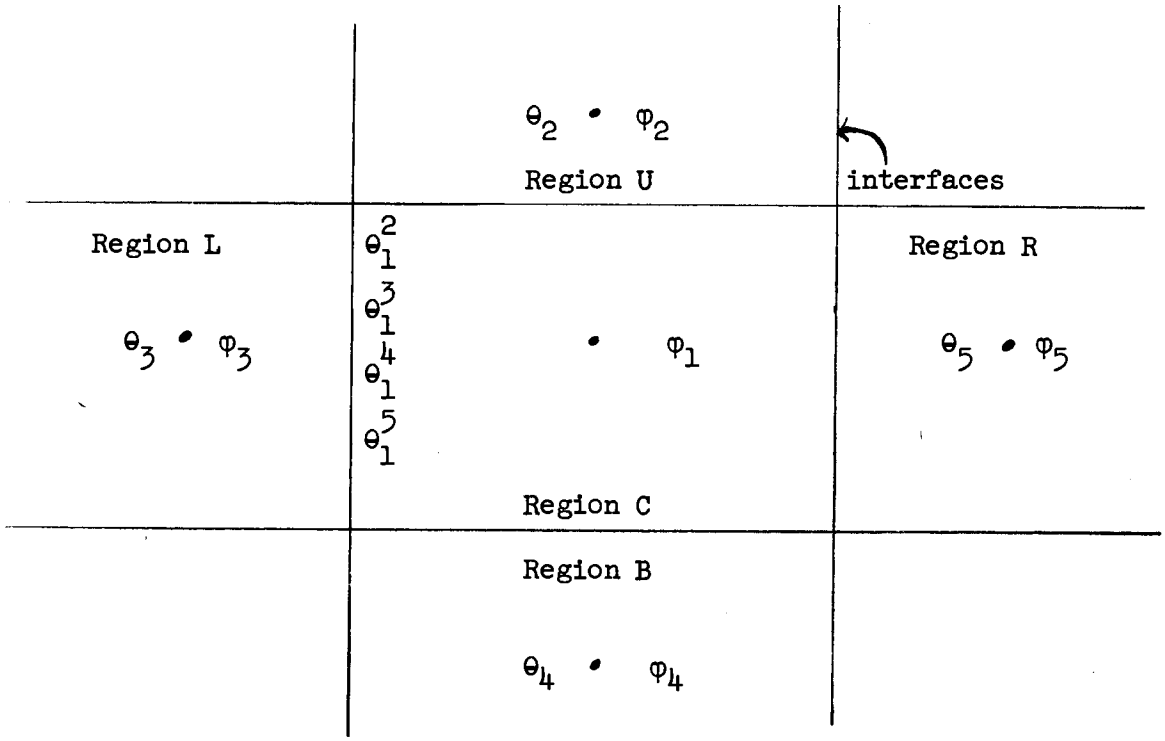

FIGURE 2

In Figure 2, the φ_i represent the true flux value and the θ_i represent fictitious values of flux. By fictitious it is meant that the flux would take on this value were there no interface. As an example, θ_2 indicates the value of the flux at mesh point 2 if there were no interface between regions U and C, and the material were all made of the type as in C. θ_1^2 indicates the flux at mesh point 1 assuming no interface and all the material of type U. $D_{f_i}^C$ indicates the diffusion coefficient associated with region C. Similar notation is used for $\Sigma_{f_i}^C$ etc. Also $D_{f_i}^U$, $D_{f_i}^B$, $D_{f_i}^L$ and $D_{f_i}^R$ are used. For the general point (11) is as follows:

$$D^C_{f_i}\left[1+\frac{\Delta r}{2r_m}\right]\theta_5 + D^C_{f_i}\left[1-\frac{\Delta r}{2r_m}\right]\theta_3 + \left\{-D^C_{f_i}\left[2+2\left(\frac{\Delta r}{\Delta z}\right)^2\right] - (\Delta r)^2\,\Sigma^C_{f_i}\right\}\varphi_1^{f_i}$$

(13)

$$+ D^C_{f_i}\left(\frac{\Delta r}{\Delta z}\right)^2\theta_2 + D^C_{f_i}\left(\frac{\Delta r}{\Delta z}\right)^2\theta_4 + \left[(\Delta r)^2\,\frac{k^C_i}{p^C_i}\,\Sigma^C_{s_i}\right]\varphi_1^{s_i} = 0 \quad .$$

Applying the given boundary conditions to the general point the following relations are obtained:

(14)
$$\begin{cases} \dfrac{\theta_2+\varphi_1}{2} = \dfrac{\theta_1^2+\varphi_2}{2} \\[2ex] D^C_{f_i}\,\dfrac{\theta_2-\varphi_1}{\Delta z} = D^U_{f_i}\,\dfrac{\varphi_2-\theta_1^2}{\Delta z} \end{cases}$$

(15)
$$\begin{cases} \dfrac{\theta_3+\varphi_1}{2} = \dfrac{\varphi_3+\theta_1^3}{2} \\[2ex] D^C_{f_i}\,\dfrac{\varphi_1-\theta_3}{\Delta r} = D^L_{f_i}\,\dfrac{\theta_1^3-\varphi_3}{\Delta r} \end{cases}$$

(16)
$$\begin{cases} \dfrac{\varphi_1+\theta_4}{2} = \dfrac{\varphi_4+\theta_1^4}{2} \\[2ex] D^C_{f_i}\,\dfrac{\varphi_1-\theta_4}{\Delta z} = D^B_{f_i}\,\dfrac{\theta_1^4-\varphi_4}{\Delta z} \end{cases}$$

$$
(17) \qquad \begin{cases} \dfrac{\varphi_1 + \theta_5}{2} = \dfrac{\varphi_5 + \theta_1^5}{2} \\[2ex] D_{f_i}^C \dfrac{\theta_5 - \varphi_1}{\Delta r} = D_{f_i}^R \dfrac{\varphi_5 - \theta_1^5}{\Delta r} \end{cases} .
$$

The above systems can be readily solved for the four fictitious values of flux, θ_2 , θ_3 , θ_4 and θ_5 .

$$
(18) \qquad \theta_2 = \frac{D_{f_i}^U (2\varphi_2 - \varphi_1) + D_{f_i}^C \varphi_1}{D_{f_i}^C + D_{f_i}^U}
$$

$$
(19) \qquad \theta_3 = \frac{D_{f_i}^L (2\varphi_3 - \varphi_1) + D_{f_i}^C \varphi_1}{D_{f_i}^C + D_{f_i}^L}
$$

$$
(20) \qquad \theta_4 = \frac{D_{f_i}^B (2\varphi_4 - \varphi_1) + D_{f_i}^C \varphi_1}{D_{f_i}^C + D_{f_i}^B}
$$

$$
(21) \qquad \theta_5 = \frac{D_{f_i}^R (2\varphi_5 - \varphi_1) + D_{f_i}^C \varphi_1}{D_{f_i}^C + D_{f_i}^R} .
$$

Substituting these values for the fictitious flux in equation (13), and rearranging gives the result,

$$\left[\frac{2D^R_{f_i}\ D^C_{f_i}\ (1+\frac{\Delta r}{2r_m})}{D^C_{f_i}+D^R_{f_i}}\right]\varphi^{f_i}_{m+1,n} + \left[\frac{2D^L_{f_i}\ D^C_{f_i}\ (1-\frac{\Delta r}{2r_m})}{D^C_{f_i}+D^L_{f_i}}\right]\varphi^{f_i}_{m-1,n}$$

$$+ \left[\frac{2D^U_{f_i}\ D^C_{f_i}\ (\frac{\Delta r}{\Delta z})^2}{D^C_{f_i}+D^U_{f_i}}\right]\varphi^{f_i}_{m,n+1} + \left[\frac{2D^B_{f_i}\ D^C_{f_i}\ (\frac{\Delta r}{\Delta z})^2}{D^C_{f_i}+D^B_{f_i}}\right]\varphi^{f_i}_{m,n-1}$$

(22) $$+ \left[\frac{(D^C_{f_i}-D^R_{f_i})\ D^C_{f_i}\ (1+\frac{\Delta r}{2r_m})}{D^C_{f_i}+D^R_{f_i}} + \frac{(D^C_{f_i}-D^L_{f_i})\ D^C_{f_i}\ (1-\frac{\Delta r}{2r_m})}{D^C_{f_i}+D^L_{f_i}}\right.$$

$$+ \left\{-2\ D^C_{f_i}\left(1+(\frac{\Delta r}{\Delta z})^2\right) - (\Delta r)^2\ \Sigma^C_{f_i}\right\} + \frac{(D^C_{f_i}-D^U_{f_i})\ D^C_{f_i}\ (\frac{\Delta r}{\Delta z})^2}{D^C_{f_i}+D^U_{f_i}}$$

$$\left.+ \frac{(D^C_{f_i}-D^B_{f_i})\ D^C_{f_i}\ (\frac{\Delta r}{\Delta z})^2}{D^C_{f_i}+D^B_{f_i}}\right]\varphi^{f_i}_{m,n} + \left[(\Delta r)^2\ \frac{k^C_i}{p^C_i}\ \Sigma^C_{s_i}\right]\varphi^{s_i}_{m,n} = 0 \quad .$$

Setting A^f equal to the coefficient of $\varphi^{f_i}_{m+1,n}$ etc., the following system can be written:

(23) $$A^f\ \varphi^f_{m+1,n} + B^f\ \varphi^f_{m-1,n} + C_f\ \varphi^f_{m,n+1} + D^f\ \varphi^f_{m,n-1} + E^f\ \varphi^f_{m,n} + F^f\ \varphi^s_{m,n} = 0\ ,$$

or in matrix notation,

$$A_f \varphi_f + F_f \varphi_s = 0 \quad . \tag{24}$$

Equation (22) is an expression for the flux at any point in the reactor as a function of the four surrounding points. In the code each point is considered to be a general point, as far as regions are concerned, and equation (22) will degenerate to the proper relation if any or all of the interfaces are removed.

In an exactly analogous manner, a general expression is determined for the slow flux. The only additional difficulty is the application of the boundary conditions which holds for the slow flux across the thin shell. The expression similar to (14) is,

$$\left\{ \begin{aligned} & \frac{\varphi_1 + \theta_2}{2} = \frac{\varphi_2 + \theta_1^2}{2} \\ & D^C_{s_i} \left(\frac{\theta_2 - \varphi_1}{\Delta z} \right) = D^U_{s_i} \left(\frac{\varphi_2 - \theta_1^2}{\Delta z} \right) + \Sigma_a t^U \left[\frac{\varphi_1 + \theta_2}{2} \right] \end{aligned} \right. \quad . \tag{25}$$

The relation which holds for the slow flux is,

$$\left[\frac{2D^R_{s_i} D^C_{s_i} (1 + \frac{\Delta r}{2r_m})}{D^C_{s_i} + D^R_{s_i} - \Sigma_a \frac{\Delta r}{2} t^R} \right] \varphi^{s_i}_{m+1,n} + \left[\frac{2D^L_{s_i} D^C_{s_i} (1 - \frac{\Delta r}{2r_m})}{D^C_{s_i} + D^L_{s_i} + \Sigma_a \frac{\Delta r}{2} t^L} \right] \varphi^{s_i}_{m-1,n} \tag{26}$$

$$+ \left[\frac{2D^U_{s_i} \; D^C_{s_i} \; (\frac{\Delta r}{\Delta z})^2}{D^C_{s_i} + D^U_{s_i} - \Sigma_a \frac{\Delta z}{2} t^U} \right] \varphi^{s_i}_{m,n+1} + \left[\frac{2D^B_{s_i} \; D^C_{s_i} \; (\frac{\Delta r}{\Delta z})^2}{D^C_{s_i} + D^B_{s_i} + \Sigma_a \frac{\Delta z}{2} t^B} \right] \varphi^{s_i}_{m,n-1}$$

$$+ \left[\frac{(D^C_{s_i} - D^R_{s_i} + \Sigma_a \frac{\Delta r}{2} t^R) \; D^C_{s_i} \; (1 + \frac{\Delta r}{2r_m})}{D^C_{s_i} + D^R_{s_i} - \Sigma_a \frac{\Delta r}{2} t^R} + \frac{(D^C_{s_i} - D^L_{s_i} - \Sigma_a \frac{\Delta r}{2} t^L) \; D^C_{s_i} (1 - \frac{\Delta r}{2r_m})}{D^C_{s_i} + D^L_{s_i} + \Sigma_a \frac{\Delta r}{2} t^L} \right.$$

$$+ \frac{(D^C_{s_i} - D^U_{s_i} + \Sigma_a \frac{\Delta z}{2} t^U) \; D^C_{s_i} \; (\frac{\Delta r}{\Delta z})^2}{D^C_{s_i} + D^U_{s_i} - \Sigma_a \frac{\Delta z}{2} t^U} + \frac{(D^C_{s_i} - D^B_{s_i} - \Sigma_a \frac{\Delta z}{2} t^B) \; D^C_{s_i} \; (\frac{\Delta r}{\Delta z})^2}{D^C_{s_i} + D^B_{s_i} + \Sigma_a \frac{\Delta z}{2} t^B}$$

$$\left. + \; D^C_{s_i} \left(-2 - 2(\frac{\Delta r}{\Delta z})^2 \right) - (\Delta r)^2 \Sigma^C_{s_i} \right] \varphi^{s_i}_{m,n} + \left[(\Delta r)^2 \; p^C_i \; \Sigma^C_{f_i} \right] \varphi^{f_i}_{m,n} = 0 \quad .$$

The system can be written as,

$$A^s \varphi^s_{m+1,n} + B^s \varphi^s_{m-1,n} + C^s \varphi^s_{m,n+1} + D^s \varphi^s_{m,n-1} + E^s \varphi^s_{m,n} + F^s \varphi^f_{m,n} = 0 \tag{27}$$

or

$$A_s \varphi^s + F_s \varphi^f = 0 \quad . \tag{28}$$

THE CODE

The code is divided into six parts, according to the logical procedure of the problem. The code considers the mesh points as being numbered according to the notation in Figure 3.

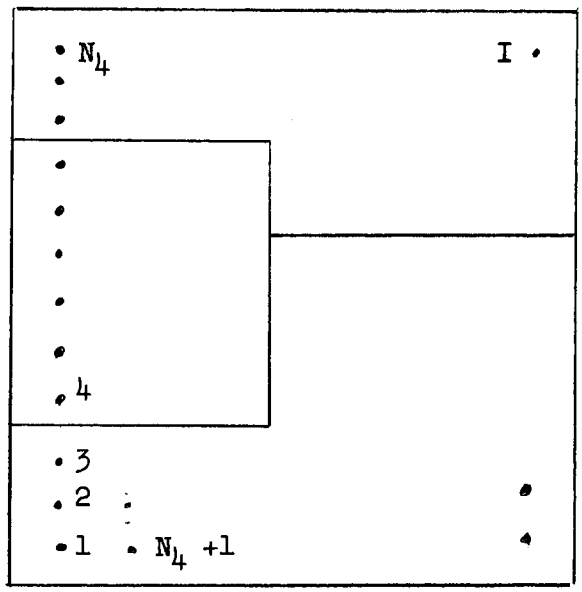

FIGURE 3

Figure 4 is a general flow chart of the complete problem.

The function of each of the six parts of the code is discussed in the following:

PART I. This part of the code computes the coefficients for the two equations (22) and (26). The fast coefficients are computed first, six at a time, for all of the mesh points starting at point number 1. The computation is done in a packed floating point (8-32) number system and the coefficients are stored on magnetic tape drive zero in this form. As soon as all of the fast

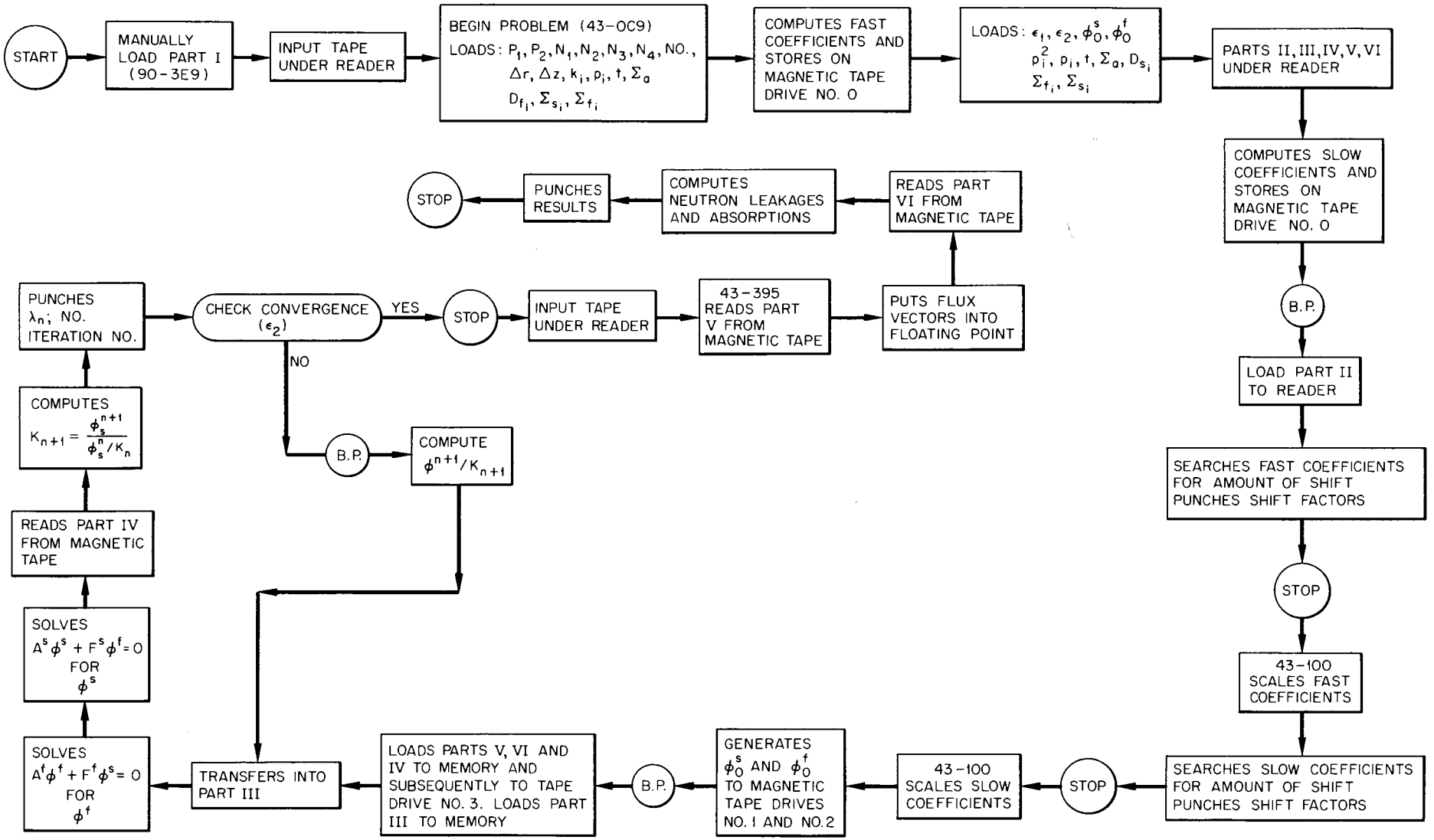
START
MANUALLY LOAD PART I (90-3E9)
INPUT TAPE UNDER READER
BEGIN PROBLEM (43-0C9)
LOADS: $P_1, P_2, N_1, N_2, N_3, N_4$, NO., $\Delta r, \Delta z, k_i, p_i, t, \Sigma_a$ $D_{f_i}, \Sigma_{s_i}, \Sigma_{f_i}$
COMPUTES FAST COEFFICIENTS AND STORES ON MAGNETIC TAPE DRIVE NO. 0
LOADS: $\epsilon_1, \epsilon_2, \phi_0^s, \phi_0^f$ $p_i^2, p_i, t, \Sigma_a, D_{s_i}$ $\Sigma_{f_i}, \Sigma_{s_i}$
PARTS II, III, IV, V, VI UNDER READER
COMPUTES SLOW COEFFICIENTS AND STORES ON MAGNETIC TAPE DRIVE NO. 0
B.P.
LOAD PART II TO READER
SEARCHES FAST COEFFICIENTS FOR AMOUNT OF SHIFT PUNCHES SHIFT FACTORS
STOP
43-100 SCALES FAST COEFFICIENTS
SEARCHES SLOW COEFFICIENTS FOR AMOUNT OF SHIFT PUNCHES SHIFT FACTORS
STOP
43-100 SCALES SLOW COEFFICIENTS
GENERATES ϕ_0^s AND ϕ_0^f TO MAGNETIC TAPE DRIVES NO. 1 AND NO. 2
B.P.
LOADS PARTS V, VI AND IV TO MEMORY AND SUBSEQUENTLY TO TAPE DRIVE NO. 3. LOADS PART III TO MEMORY
TRANSFERS INTO PART III
SOLVES $A^f\phi^f + F^f\phi^s = 0$ FOR ϕ^f
SOLVES $A^s\phi^s + F^s\phi^f = 0$ FOR ϕ^s
READS PART IV FROM MAGNETIC TAPE
COMPUTES $K_{n+1} = \frac{\phi_s^{n+1}}{\phi_s^n / K_n}$
PUNCHES λ_n; NO. ITERATION NO.
CHECK CONVERGENCE (ϵ_2)
YES
NO
STOP
INPUT TAPE UNDER READER
43-395 READS PART V FROM MAGNETIC TAPE
PUTS FLUX VECTORS INTO FLOATING POINT
READS PART VI FROM MAGNETIC TAPE
COMPUTES NEUTRON LEAKAGES AND ABSORPTIONS
PUNCHES RESULTS
STOP
B.P.
COMPUTE ϕ^{n+1}/K_{n+1}

FIGURE 4

coefficients are computed, the code then reads into the memory the necessary information from the parameter tape and readjusts itself so as to compute the slow coefficients. The slow coefficients are stored immediately following the fast coefficients on tape drive zero and are placed on the tape in exactly the same manner as the fast coefficients.

It is in this part of the code that all of the boundary conditions are imposed and all of the controls for the geometry are set up.

The storage requirements for Part I are as follows:

000 - 0B4	---	8-32 Packed Floating Point System
0B5 - 225	---	Part I Code
226 - 3A5	---	Coeff. as generated
3A6 - 3E8	---	Temporary storage and input data
3E9 - 3FF	---	Program constants.

PART II. Since, as has already been stated, Part I incorporates the geometry requirements of the problem there results only a mathematical system to solve. This can be done in a fixed point number system and the only scaling necessary is the initial scaling of the elements of the matrices A_f , A_s , F_f and F_s . The code first searches for the largest (in magnitude) element in the two matrices A_f and F_f . At this stage the necessary scaling, to make each element a fixed point machine number, is punched (in binary) along with the maximum loss of significance in any one element. There is a coded stop following this punch-out which affords an opportunity to examine the scale factors and to make sure there is no extreme loss of significance. When operation is resumed each of the fast coefficients will be scaled and transformed into a fixed point

number. This is done in such a way that a shift of 8 binary bits can be affected without loss of any of the 32 bits of significant information in a particular element. Experience has shown that in most cases none of the 32 bits of significant information is lost.

The code will perform the same operation as indicated above on the slow coefficients as soon as the scaling of the fast coefficients is complete.

Part II also generates the flat source φ_o^s and the first estimate of the fast flux φ_o^f to tape drives No. 2 and No. 1 respectively. This part of the code also loads Part IV, V and VI to the memory and subsequently to magnetic tape drive No. 3. Part III is loaded into the memory and control is transferred to the beginning of Part III.

The storage requirements of Part II are as follows:

000 - 0B4	---	Packed 8-32 Floating Point System
0B5 - 184	---	Part II code
185 - 224	---	Free
225 - 3A5	---	Coefficients for Scaling
3A6 - 3E8	---	Temporaries
3E9 - 3FF	---	Constants.

PART III. This part of the code, the so-called equation solver, solves the system

$$A\varphi + D = 0 \quad .$$

The method used is that of Richardson with the recursive formula as indicated earlier. At the present time the relaxation factors, β_n , are built into the code. A better

rate of convergence can be obtained by varying these depending upon certain properties of the matrices A_f and A_s. It was not deemed advisable to include this as a variable part of the code at the present time.

The number of equations that can be handled by Part III is arbitrary (the one limitation is that there shall not be more than 39 rows in the mesh; the number of columns is unlimited). All of the information concerning the size of the system to be solved is supplied by Part II from the input data.

Upon entry into Part III, the system

$$A_f \varphi^f + F_f \varphi^s = 0$$

is solved for φ^f, using the φ^s presently stored on magnetic tape as given. The condition for convergence of this system is set by the input number ϵ_1. Since in the beginning a flat source is generated it has been found advisable to begin with a very lax ϵ_1 and to make this successively more strict on subsequent passes through the major computing loop. This scaling of ϵ_1 is built into Part IV of the code. As soon as the vector φ^f is found to the desired degree of accuracy, the code readjusts itself to solve the system,

$$A_s \varphi^s + F_s \varphi^f = 0$$

for φ^s using the new φ^f.

There are two coded stops in Part III. These are stops which are set in the code to check for overflow in computing successive flux values. In case of stops at these stages it is necessary to scale the flux vectors.

Part IV is automatically read from magnetic tape drive No. 3 and control is transferred to Part IV. The storage requirements of Part III are,

000 - 0DF	---	Part III code
0E0 - 260	---	Storage for fast (slow) coeff.
261 - 2E0	---	Storage for fast (slow) flux values
2E1 - 360	---	Storage for slow (fast) flux values
361 - 3AE	---	Temporaries
3AF - 3FF	---	Temporaries and constants.

PART IV. The first function of this part of the code is to compute K_{n+1}. This is done by dividing the vector sum of φ_s^{n+1} by the vector sum of φ_s^n/K_n. A check is then made to determine if convergence to K is complete. The requirement here is that $\left| K_{n+1} - K_n \right| < \epsilon_2$, where ϵ_2 is an input number. The quantities K_{n+1} (in two words, an integer and a fractional part), the number of iterations required to solve the system $A_s \varphi^s + F_s \varphi^f = 0$, and the problem number are punched. If convergence to K is complete the code stops, otherwise the vector φ_s^{n+1} is normalized to φ_s^{n+1}/K_{n+1} and Part III is reentered. The storage requirements for Part IV are as follows:

300 - 39F	---	Part IV code
All others as in Part III.		

PART V. The lone function of this part of the code is to transform the elements of the two flux vectors from fixed point numbers to floating point numbers. Prior to the transfer command which is given to enter Part V, the input

tape should again be placed under the reader. The storage requirements of Part V are,

300 - 33F --- Part V.

PART VI. The function of this part of the code is to perform the numerical integration to evaluate the integrals (8) and (9). The parameters are again loaded from the input tape to be used in the calculation of these integrals. In all parts of the calculation up to this point the overall dimensions of the reactor (R_1 and H) have included the extrapolation distance, with no reference to the true reactor dimensions. In computing the neutron absorptions and leakages this must be considered.

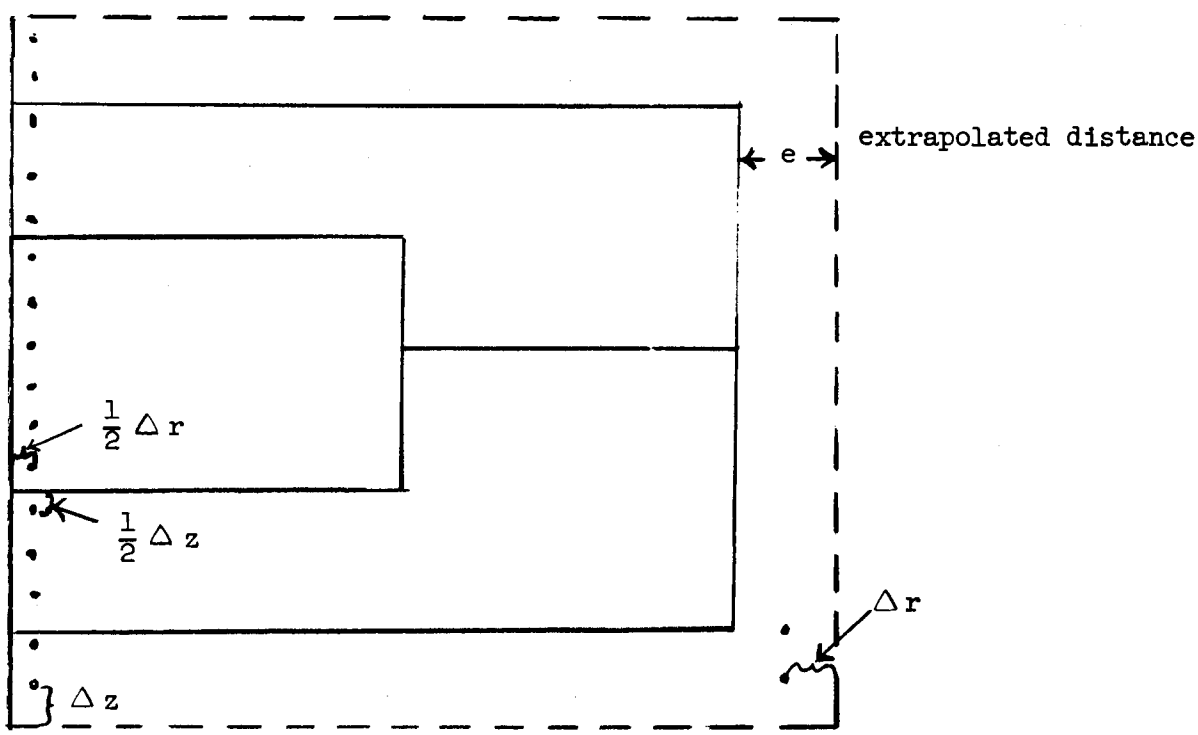

FIGURE 5

The input numbers which take into consideration the extrapolated distance are A_r , M_1 and M_2 . The number $A_r = 2(R_1 - e)(H - 2e)$ is simply the lateral area (with the factor π omitted since this is a common factor in all of the areas and volumes) of the cylinder and is used in computing leakages in the radial direction. M_1 is the number of mesh points per column which lie between the extrapolated boundary and the true boundary (on one end of the reactor), and M_2 is the number per row. These are control numbers used so that the absorptions in a given region will not include points in the extrapolated area and so that leakages will be computed from the first flux value immediately outside the true boundary. All of the computation in this part of the code is done in floating point. The output (in decimal floating point) is the value of the 16 integrals indicated in (8) and (9). The storage requirement of Part VI is

000 - 0B4	---	Packed Floating Point
0B5 - 1F1	---	Part VI code
1F2 - 1FF	---	Free
200 - 280	---	128 Fast Flux Values
281 - 28F	---	128 Slow Flux Values
300 - 37F	---	Free
380 - 3BF	---	Temporaries
3C0 - 3FF	---	Constants.

PREPARATION OF INPUT DATA

Preparation of the input tape can perhaps be best illustrated by use of an example. With reference to Figures 1 and 5, suppose the following dimensions

and material data are given:

Dimensions	Region 1	Region 2
R_o = 70 cm.	D_f = 1.87	D_f = 1.57
R_1 = 150 cm.	D_s = 1.43	D_s = 1.23
a = 80 cm.	p = 1.00	p = 1.00
b = 140 cm.	$\Sigma_f = 8.68 \times 10^{-3}$	$\Sigma_f = 7.62 \times 10^{-2}$
c = 200 cm.	$\Sigma_s = 2.91627 \times 10^{-3}$	$\Sigma_s = 4.45 \times 10^{-4}$
H = 300 cm.	k = 1.499	k = 0
e = 15 cm.		

Region 3	Shell
D_f = 1.25	t = 1.37
D_s = 1.13	$\Sigma_a = 6.417 \times 10^{-3}$
p = 0.40	
$\Sigma_f = 6.17 \times 10^{-3}$	
$\Sigma_s = 3.119 \times 10^{-2}$	
k = .159996	Extrapolated distance = 15 cm.

The first thing to establish is the mesh to be used. It is not necessary to have the same number of mesh points in the rows as in the columns, nor is it necessary to have Δr equal Δz. It has been found that in the average problem a mesh in the order of 20 x 20 is sufficient. It is not possible in many cases to set up the mesh in such a way that the exact dimensions will be realized; however, any variation (which can be made quite small through the proper choice of a mesh)

can be incorporated into the extrapolation distance. A suitable mesh for the example might be as follows:

$\Delta r = 7.0$ cm.

$\Delta z = 15.0$ cm.

$P_1 = 10$

$P_2 = 21$ (note that $(21.5)(\Delta r) = 150.5$. Inspection of Figure 5 will indicate the reason for the factor 21.5)

$N_1 = 5 \quad \left[(5.5)(\Delta z) = 82.5\right]$

$N_2 = 9 \quad \left[(9.5)(\Delta z) = 142.5\right]$

$N_3 = 13 \quad \left[(13.5)(\Delta z) = 202.5\right]$

$N_4 = 19 \quad \left[(20)(\Delta z) = 300.0\right]$

The dimensions of the core have been left unchanged, but the 2.5 cm. have been incorporated into the extrapolation distance.

$A_r = 7.29 \times 10^4$

$M_1 = 1$

$M_2 = 2$.

Experience has shown that .001 is a good beginning value for φ_o^s , φ_o^f and ϵ_1 . ϵ_2 can of course vary with the degree of accuracy one wishes for K , but .0005

is considered good enough to give K correct to 3 or 4 figures.

The complete input tape and the manner in which it should be typed is indicated in a table below. The following should be noted:

1. The quantities P_1 , P_2 , N_1 , N_2 , N_3 , N_4 are typed as decimal integers.

2. The problem number can be any ten decimal digits and will be punched out exactly as it appears on the input tape.

3. The quantities Δr through the second N_4 are typed in a decimal floating point form to be used with Oracle Subroutine No. 035. These numbers are of the form

$$\overset{0}{F}\ \text{xxx xxx}\ \overset{0}{F}\ \text{yy}$$

representing the number $\pm x \cdot 10^{\pm y}$. The exponent range is limited to $0 \leqslant y \leqslant 37$ and the fractional part to $.1 \leqslant x \leqslant .999999$. The representation of zero is 00 000 00 000 .

4. The quantities M_1 and M_2 are typed as decimal integers.

00 000 00 010	P_1
00 000 00 021	P_2
00 000 00 005	N_1
00 000 00 009	N_2
00 000 00 013	N_3
00 000 00 019	N_4
01 234 56 789	#
	Double Space

07 000 00 001	Δr
01 500 00 002	Δz
01 499 00 001	k_1
00 000 00 000	k_2
01 599 96 000	k_3
01 000 00 001	p_1
01 000 00 001	p_2
04 000 00 000	p_3
00 000 00 000	zero
00 000 00 000	zero
01 870 00 001	D_{f_1}
01 570 00 001	D_{f_2}
01 250 00 001	D_{f_3}
02 916 27 F02	Σ_{s_1}
04 450 00 F03	Σ_{s_2}
03 119 00 F01	Σ_{s_3}
08 680 00 F02	Σ_{f_1}
07 620 00 F01	Σ_{f_2}
06 170 00 F02	Σ_{f_3}
01 000 00 F02	ϵ_1
05 000 00 F03	ϵ_2
01 000 00 F02	φ_o^s
01 000 00 F02	φ_o^f
01 000 00 001	p_1^2

01 000 00 001	p_2^2
01 600 00 000	p_3^2
01 000 00 000	p_1
01 000 00 000	p_2
04 000 00 000	p_3
01 370 00 001	t
06 417 00 F02	Σ_a
01 430 00 001	D_{s_1}
01 230 00 001	D_{s_2}
01 130 00 001	D_{s_3}
08 680 00 F02	Σ_{f_1}
07 620 00 F01	Σ_{f_2}
06 170 00 F02	Σ_{f_3}
02 916 27 F02	Σ_{s_1}
04 450 00 F03	Σ_{s_2}
03 119 00 F01	Σ_{s_3}
07 290 00 005	Ar
01 900 00 002	N_4
00 000 00 001	M_1
00 000 00 002	M_2
	Double Space

THE PROBLEM OUTPUT

The first output from the code will be the scaling factors. These are four numbers punched for each of the two groups. They are,

(a) The amount by which each element of the matrices A_f and F_f (or A_s and F_s in case of the slow group) must be scaled. This is punched out in starred hexadecimal form.

(b) The amount of shift that will occur in the worst case. In other words, the maximum loss of significance. This again is punched in starred hexadecimal form.

(c) The largest element in the matrices A_f and F_f (or A_s and F_s) punched in decimal floating point form.

(d) The smallest element in the matrices A_f and F_f (or A_s and F_s).

The next output information is the successive values of K. A new K is punched each time an iteration is completed, that is, each time that Part IV of the code is entered. The K is punched in the form of two words, an integer part and a fractional part. Along with each K there is also punched the number of iterations that were necessary in solving the system $A_s \varphi^s + D = 0$ at the preceding stage. The problem number is also punched.

The final output information is the value of each of the 16 integrals. These are punched in floating decimal form. They are arranged as follows:

$A^T_{s_1}$	$A^T_{f_1}$	$A^T_{s_2}$	$A^T_{f_2}$
$A^T_{s_3}$	$A^T_{f_3}$	A^T_2 (shell)	A^T_3 (shell)
$L^z_{s_2}$	$L^z_{f_2}$	$L^z_{s_3}$	$L^z_{f_3}$
$L^r_{s_2}$	$L^r_{f_2}$	$L^r_{s_3}$	$L^r_{f_3}$.

LIMITATIONS OF THE CODE

There are certain limitations to the code that may not have been specifically mentioned previously. The following is an attempt to summarize the limitations and in so doing point out the generalities:

a. $N_4 \leqslant 39$

b. $N_1 \geqslant 2$

c. $N_3 - N_1 \geqslant 3$

d. $N_4 - N_3 \geqslant 3$

e. $N_2 - N_1 \geqslant 2$

f. $N_3 - N_2 \geqslant 2$

g. $M_1 \geqslant 1$, $M_2 \geqslant 1$

h. $P_1 \geqslant 1$

i. $P_2 - P_1 \geqslant 2$.

OPERATING PROCEDURE

The complete operating procedure can be summarized as follows:

a. Load Part I of the code. 90 - 3E9.

b. Parameter tape under reader. 43 - 0C9. B.P. off.

c. As soon as parameter tape is read in, place Part II of code under reader.

d. At punch out of scale information for fast group, 43 - 100.

e. At punch out of scale information for slow group, 43 - 100.

f. There are no other coded stops (except for overflow) until convergence to K is complete.

g. The machine will stop when K has been calculated to the desired degree of accuracy. At this time if it is desired to calculate the neutron balance, place the parameter tape again under the reader and 43 - 395.

h. The machine will stop when all integrals have been calculated and punched.

The stops which are coded into the problem are as follows:

Right 0F2 --- After punch out of scale information. (Part II)

Right 049 --- Overflow stop. (Part III)

Left 0D8 --- Overflow stop. (Part III)

Right 326 --- Stop when convergence to K is complete. (Part IV)

Left 1F1 --- Stop when integrals have been punched. (Part VI)

Certain breakpoints for optional stopping are provided as follows:

Left 220 --- Stop before loading Part II. (Part I)

Right 328 --- Stop after punch out of K . (Part IV)

Certain precautions have been taken to prevent loss of information in case of failure to read on one of the magnetic tapes. All of the information that is computed is duplicated on other parts of the magnetic tape. The coefficients

generated by Part I are duplicated on Drive No. 3 immediately after they have been scaled and transformed into fixed point numbers. This is done block for block starting with block 20 on Drive No. 3. The correspondence is as follows:

Drive No. 0	Drive No. 3
Block 1	Block 20
Block 2	Block 21, etc.

The flux vectors are duplicated after each complete iteration. This is done by Part IV of the code as follows:

FAST FLUX

Drive No. 1	Drive No. 2
Block 1	Block 20
Block 2	Block 21, etc.

SLOW FLUX

Drive No. 2	Drive No. 1
Block 1	Block 20
Block 2	Block 21, etc.

In case a block of the magnetic tape will not read it is always possible to find information duplicated and the problem can be corrected without a great loss in computing time.

The time necessary to run a typical problem (a mesh of 400 points) is in the neighborhood of one hour. The time varies almost linearly with the number of mesh points used.

NOMENCLATURE

a -- Distance from $z = 0$ to first horizontal interface. (cm.)

b -- Distance from $z = 0$ to second horizontal interface. (cm.)

c -- Distance from $z = 0$ to third horizontal interface. (cm.)

e -- Extrapolated distance. (cm.)

H -- Height of reactor including extrapolated distance. (cm.)

R_o -- Radius of region 1. (cm.)

R_1 -- Radius of reactor including extrapolated distance. (cm.)

t -- Thickness of shell around region 1. (cm.)

r_m -- Distance from $r = 0$ to variable mesh point in r-direction. (cm.)

A_r -- Lateral area of reactor (true dimensions). (cm.)

Δr -- Increment in r-direction. (cm.)

Δz -- Increment in z-direction. (cm.)

N_1 -- Number of mesh points in column from $z = 0$ to $z = a$.

N_2 -- Number of mesh points in column from $z = 0$ to $z = b$.

N_3 -- Number of mesh points in column from $z = 0$ to $z = c$.

N_4 -- Number of mesh points in column from $z = 0$ to $z = H$.

P_1 -- Number of mesh points in row from $r = 0$ to $r = R_o$.

P_2 -- Number of mesh points in row from $r = 0$ to $r = R_1$.

I -- Number of points in entire mesh.

M_1 -- Number of mesh points in column between true dimensions and extrapolated dimension.

M_2 -- Number of mesh points in row between true dimension and extrapolated dimension.

ϵ_1 -- Convergence criteria on $A\varphi + D = 0$.

ϵ_2 -- Convergence criteria on K .

φ -- Flux value.

θ -- Fictitious flux value.

K -- Effective multiplication constant for reactor system.

Σ_a -- Thermal macroscopic absorption cross section of thin shell (cm^{-1})

Σ_{f_i} -- Fast macroscopic absorption cross section of region i . (D/τ) (cm^{-1}).

Σ_{s_i} -- Thermal macroscopic absorption cross section of region i. (cm^{-1}).

τ_i -- Fermi age for region i. (cm^2).

D_{f_i} -- Fast diffusion coefficient of region i. (cm.)

D_{s_i} -- Thermal diffusion coefficient of region i. (cm.)

p_i -- Resonance escape probability in region i.

k_i -- Infinite multiplication constant for region i.

CLB:ih